Alexandria University
Faculty of Engineering
Computer and Systems Engineering Department
Object Oriented Programming

Vector Drawing Report

Submitted by : Moustafa Mahmoud & Salma Ahmed

11/3/2016

Introduction:

The presented Vector Drawing Application offers several options including:

1. Drawing five shapes by which are rectangle, triangle, ellipse, circle and one additional shape which is loaded as a plug-in, the square.
2. Moving the drawn shapes within the drawing grid.
3. Resizing any of the shapes.
4. Selecting a certain outline color and fill color.
5. Undo and redo actions.
6. Deleting any drawn shape.
7. Editing a previously set outline or fill color.
8. Creating a new grid using the new option.
9. Changing the stroke thickness.
10. Saving and loading any of the drawings where two file saving formats are supported, XML&Json.

Extra features supported by our application:

11. Saving your painted drawing as a jpg image to be able to view later
12. A keyboard listener is added in order to allow the user to do some actions like undo, redo save and load using the keyboard.

User Guide:

1. **Modes Menu:**
   The user toggles between the following 3 modes using the modes menu:
   a. **Draw:**
      Drawing any of the supported shapes is done through pressing the button referring to the required shape, and then start drawing the shape onto the grid by dragging till a shape with the required dimensions is drawn.
   b. **Move:**
      First, a certain drawn shape is selected. Then the move mode is chosen from the modes menu. After that, the user is free to move the shape anywhere along the drawing grid
   c. **Resize:**

      First, a certain drawn shape is selected. Then the Resize mode is chosen from the modes menu. After that, the user is allowed to resize the shape by dragging it in any of the directions.

2. **Buttons Panel:**
   - Featuring all the supported shapes buttons and a button for the shape "Square" which appears only when the square plug-in is loaded.
   - Edit button which allows you to choose between two options: Editing the outline color or the fill color.
   - Delete button which deletes the selected shape.
   - Undo button which on click undoes the last action done by the user whether it is drawing, moving, resizing or coloring a shape till there is no undone actions left so a message indicating this appears to the user.

- Redo button which on click redoes the undone action by the user till there is no redo history left so a warning message appears.
- Save button on click a file chooser appears where you can choose where to save the XML or Json file and if another file extension is entered error message appears.
- Load button on click a file chooser appears which allows the user to choose the location to load the file from then the grid is redrawn with the loaded shapes.

3. **<u>Outline Color Button:</u>**
   Allows the user to set the outline color of the shapes to be drawn and which is set to black by default.
4. **<u>Fill Color Button:</u>**
   Allows the user to set the fill color of the shapes to be drawn and which is set to white by default.
5. **<u>File Menu:</u>**
   Featuring several options which are:
   - New: creates a new blank drawing grid.
   - Save
   - Save image as jpg: which allows the user save his drawing as a jpg image to view later.
   - Load
   - Load plug –in: which allows the user to add shapes plug-ins to the program while runtime.
   - Exit: allows the user to close the program but the user first chooses either to save or not to save the drawings before closing.
6. **<u>Stroke Menu:</u>**
   Featuring three available thicknesses for the stroke:
   - Thin: This is the default stroke.
   - Medium.
   - Thick.

Design Description:

The project is divided into several packages containing several classes as stated below:

1. Package Shapes:

It includes an abstract class “shape” and the shapes classes “Rectangle, Square, Ellipse, Circle, Triangle”.

a. An abstract class called “shape” includes all the common methods supported by the shapes where each shape inherits from this class.
b. Some shapes are extended from other ones for example the square is extended from the rectangle and the circle is extended from the ellipse.
c. Each shape class includes a constructor for the shape for defining its attributes and the following methods:
    - Setters and getters for the shape attributes.
    - Draw which draws a certain shape according to its attributes.
    - PointInShape which determines if a certain point is contained within the shape to help in selecting it.
    - createXMl which sends the attributes of the shape to an XML writer class.

- createJson which sends the attributes of the shape to a Jsonbwriter class.

2. Package TempShapes:

It includes the following classes:

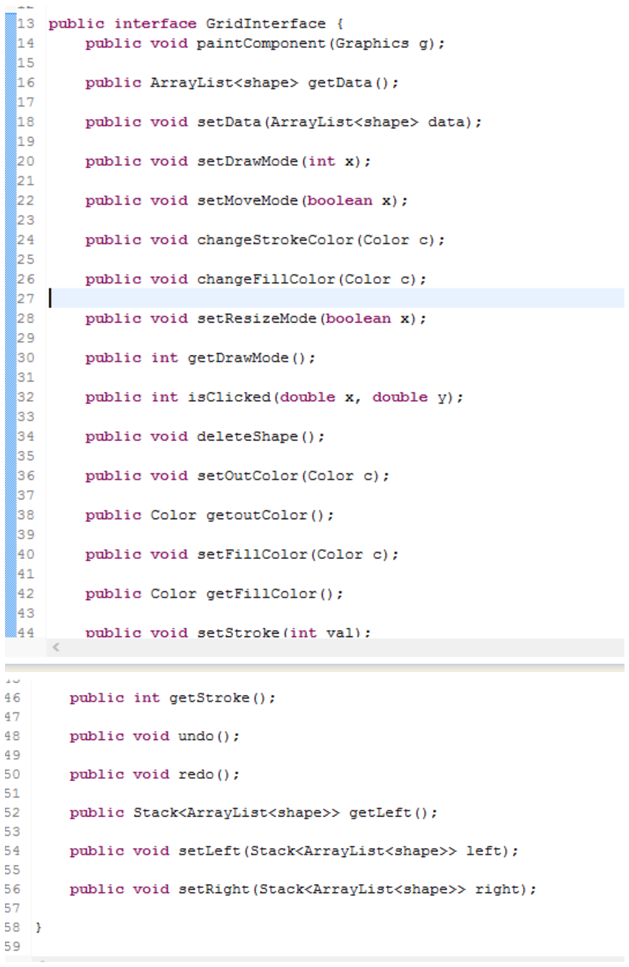
a. ChangeColor
b. MoveShape
c. RseizeShape
d. WhileDraw
e. WhileMove
f. WhileResize

Each of these classes makes sure that if a change is done to the shape it is applied and the previous state of the shape is also maintained without manipulating the previous reference to the shape.

3. Package DrawTools:

It includes the following classes:

1. GridInterface:
   It defines an interface for the painting grid including the following methods:



```
public interface GridInterface {
    public void paintComponent(Graphics g);

    public ArrayList<shape> getData();

    public void setData(ArrayList<shape> data);

    public void setDrawMode(int x);

    public void setMoveMode(boolean x);

    public void changeStrokeColor(Color c);

    public void changeFillColor(Color c);

    public void setResizeMode(boolean x);

    public int getDrawMode();

    public int isClicked(double x, double y);

    public void deleteShape();

    public void setOutColor(Color c);

    public Color getoutColor();

    public void setFillColor(Color c);

    public Color getFillColor();

    public void setStroke(int val);
```

```
    public int getStroke();

    public void undo();

    public void redo();

    public Stack<ArrayList<shape>> getLeft();

    public void setLeft(Stack<ArrayList<shape>> left);

    public void setRight(Stack<ArrayList<shape>> right);

}
```

2. PaintGrid:
   It performs the core logic of the functions in the above interface
3. GUI_paint:
   It contains the graphical user interface code used to set its design and the calls for every button or action within the program.

4. Package Files:

It includes the following classes related to dealing with files saving , loading and loading plug-ins:

a. FileTools:
   a class designed to take a file path and then gets the extension of this file which helps in determining whether it is an xml or Json file or even a jar.
b. JsonWriter:
   a class responsible for creating a json array for saving the arraylist of shapes drawn where for each shape a json object is created with its attributes then it is written to a certain file.
c. XmlWriter:

   a class responsible for creating a DOM document for saving the arraylist of shapes drawn where for each shape text nodes are created with its attributes then it is written to a certain file.

d. JsonReader

   It is responsible for parsing the json array written in the saved file and getting its attrinbutes in order to redraw the shapes after loading

e. XmlReader:

   It is responsible for parsing the text nodes written in the saved file according to their tag name and getting its attributes in order to redraw the shapes after loading

f. loadClasses: responsible for loading the plug-ins or as called dynamically loaded classes.

Design Decisions:

1. In order to move or resize a certain shape the user must choose which mode is he in order to perform that action.
2. Resizing is to be done in a user friendly way but without the use of small rectangular boxes.
3. Whenever the user loads a previously saved drawing, the undone and redone actions history is not saved so after loading the user is not allowed to traverse the history.

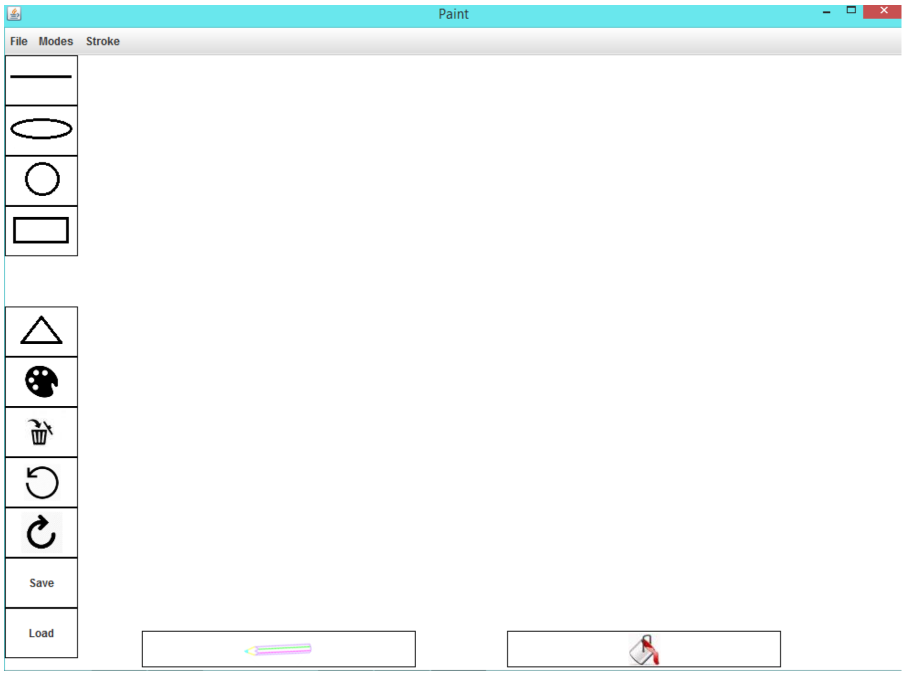
4. We decided that both the square and circle classes should extend from the rectangle and ellipse classes respectively as they have common properties.

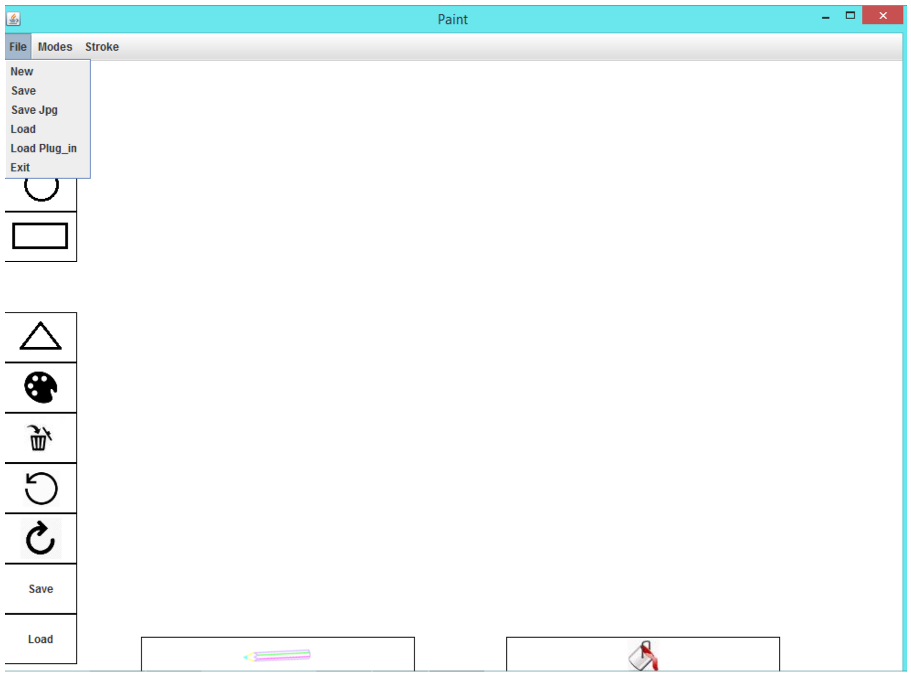
5.In the undo and redo actions we used a stack of array lists which takes a memory of order (n^2) although we could have reduced it to order(n) which allows saving history up to 1000000 operation but it is common sense that the user is not going to operate more than 1000 operations so the order n^2 is enough.

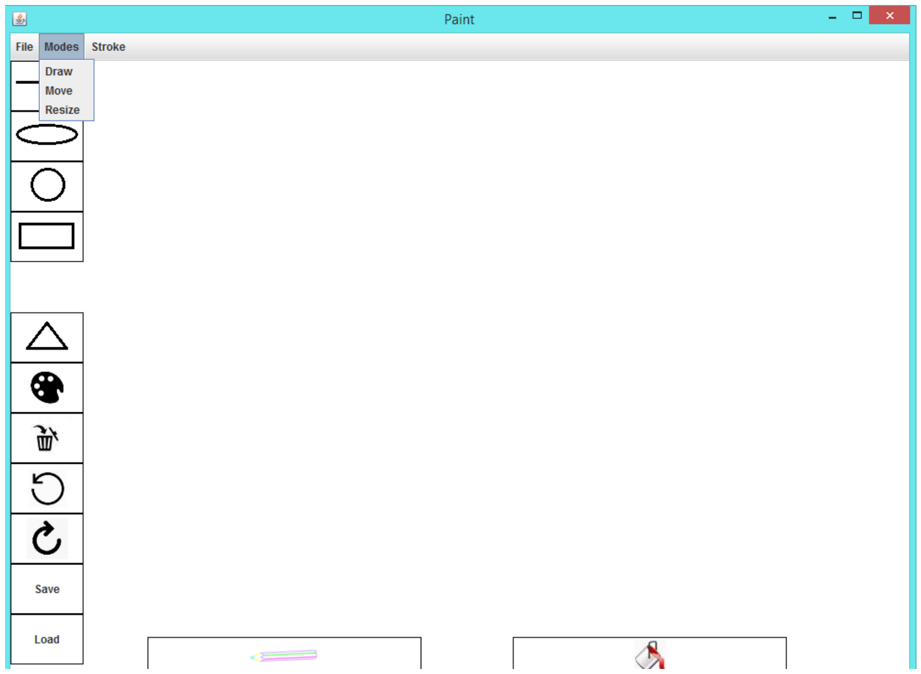
6.the code is divided into packages each containing certain classes which support a certain functionality to make the code more readable.

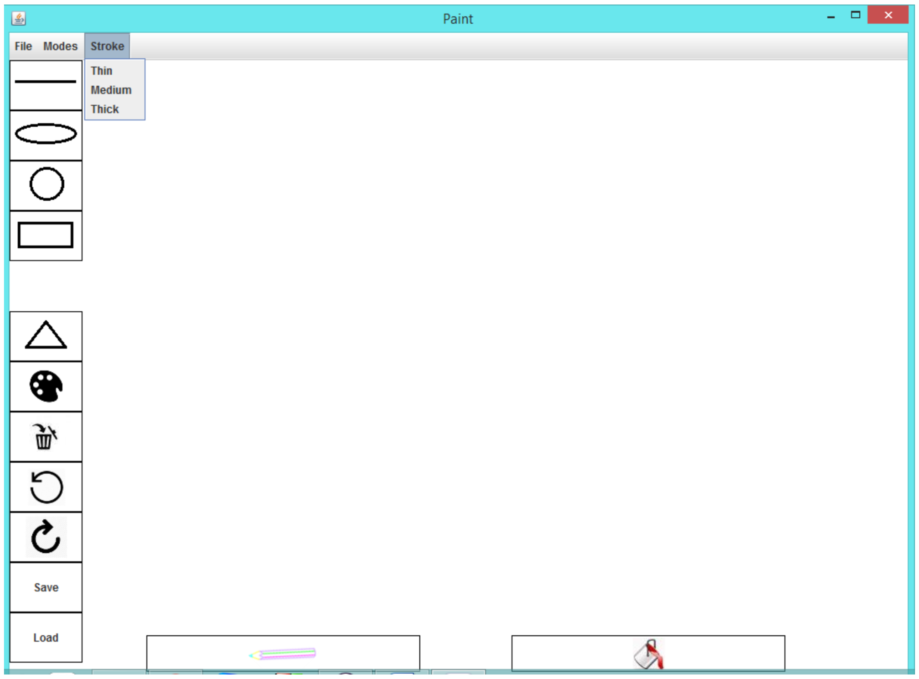
8.We exported the square class as a plug in but in the same our program could have more plug ins added to it in the future so it can be developed.

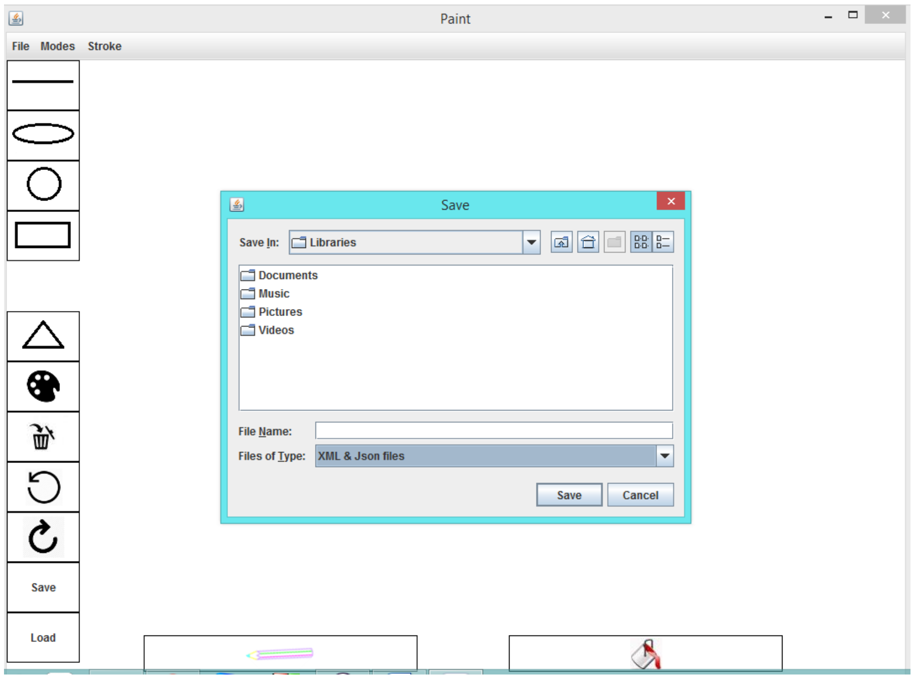
GUI Snapshots:

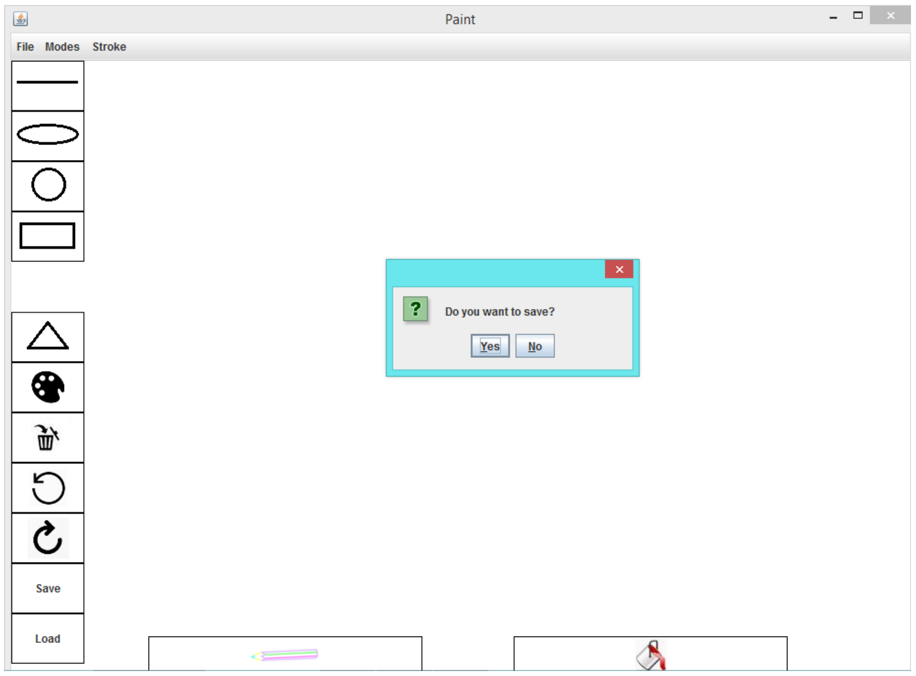



Paint
File
Modes
Stroke
New
Save
Save Jpg
Load
Load Plug_in
Exit
Save
Load

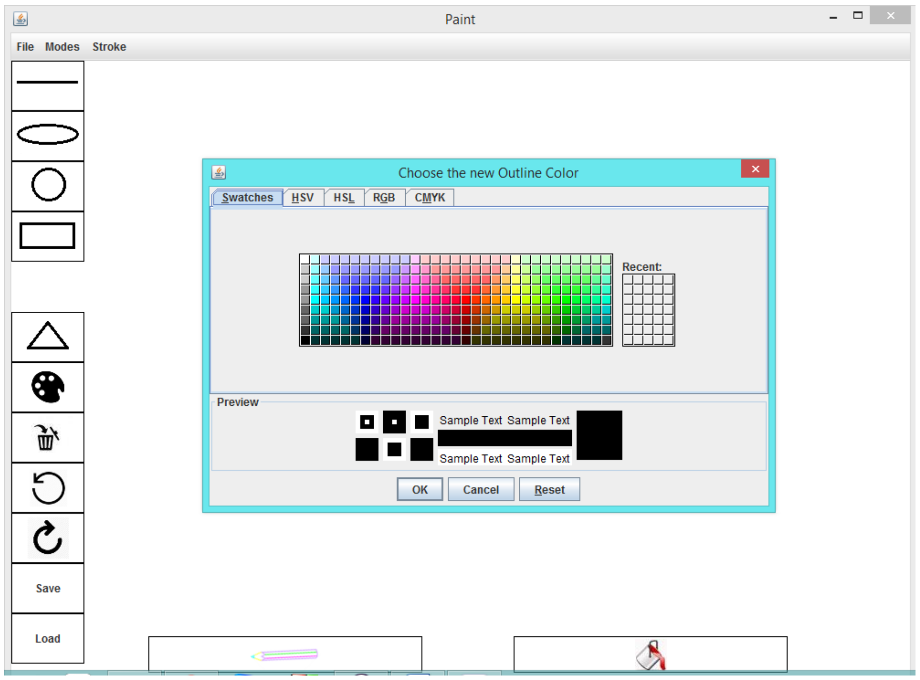

Paint
File
Modes
Stroke
Draw
Move
Resize
Save
Load


Paint
File
Modes
Stroke
Thin
Medium
Thick
Save
Load


Paint
File
Modes
Stroke
Save
Save In:
Libraries
Documents
Music
Pictures
Videos
File Name:
Files of Type:
XML & Json files
Save
Cancel
Save
Load



Do you want to save?


Paint
File Modes Stroke
Save
Load
Choose the new Outline Color
Swatches HSV HSL RGB CMYK
Recent:
Preview
Sample Text Sample Text
Sample Text Sample Text
OK Cancel Reset